

Unit 1: PythonによるTI-Innovator™ Hubのスタート

Skill Builder 3: 入力とサウンド

このレッスンでは、TI-Innovator Hubでサウンドを作成し、変数を使って値を保存・取得する2つの方法について学習します。

目標

- TI-Innovator Hubでサウンドを生成
- 変数を使って入力値を格納
- **input()**関数と**int()**関数の使用
- 音符の再生

TI-Innovator Hubには、SOUND(音)と呼ばれる小さなスピーカーがあります。スピーカーはHubの下部にあります。アンプがないので音量は非常に小さいものです。教室がうるさくなりすぎないように設計されています。



1. Hub Projectテンプレートを使って新規のPythonプロジェクトを開始します。

新規のドキュメントを開始するか(**home > New(ホーム>新規)**), 現在のドキュメントにページを追加する(**ctrl+doc**)ことができます。 **Add Python > New...(Python追加>新規...)**を選択します。必ずHub Projectテンプレートを選択してください。

```

1.4 1.5 1.6 *Unit 1 P...und RAD
*U1SB3.py 9/17
#Unit1 SB3
#-----
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#-----

```

2. soundを作成するため値を取得するには、2つの入力ステートメントを記述します。これらの値は、2つの変数f, tに格納されます。以下のように記述します。

f = input("Frequency? ")
t = input("Time? ")



これをステップバイステップで行います。



3. コメント行#=====の下の空白行に文字fと=記号を入力します。

f =

次のステップのためカーソルを=記号の右側に置きます。

```

1.4 1.5 1.6 *Unit 1 P...und RAD 10/18
# Unit1 SB3
#=====
from ti_hub import *
from math import *
from time import sleep
from tiplotlib import text_at,cls
from ti_system import get_key
#=====
f=|

```

4. つぎに、**menu > Built-ins > I/O**(メニュー>組み込み>I/O)から**input()**関数を選択します。

```

1 Actions
2 Run
3 Edit
4 Built-ins
5 Math
6 Random
7 TI PlotLib
8 TI
9 TI
A M
var B Va
1 Functions
2 Control
3 Ops
4 Lists
5 Type
6 I/O
1 print()
2 input()
3 eval()
4 .format()

```

5. メニューから**input()**を選択すると、=記号の後に**input()**という単語がプログラムに貼り付けられます。括弧内に、この入力ステートメントのプロンプト(入力要請)を入力します。

グラフ電卓では、引用符(")は**ctrl+x**(かけ算キー)です。大文字の場合は、**shift**キーを使います。キーパッドの文字Gの横にある句読点キー(,)の疑問符(?)を見つけます。必ず閉じの引用符も入力してください。

次の行で、変数tの入力ステートメントを作成します。

```

1.4 1.5 1.6 *Unit 1 P...und RAD 10/19
# Unit1 SB3
#=====
from ti_hub import *
from math import *
from time import sleepinput()
from tiplotlib import text_at,cls
from ti_system import get_key
#=====
f=input("Frequency? ")|

```

6. 2つの入力ステートメントの下で、**sound.tone()**関数を次から取得します。

menu > TI Hub > Hub Built-in Devices > Sound Output
(メニュー>TI Hub>Hub内蔵デバイス>サウンド出力)

次のステートメントが表示されます。

sound.tone(frequency, time)

frequency (周波数) の代わりに変数fを入力し、
timeの代わりに変数tを入力します。

```

1.5 1.6 1.7 *Unit 1 P...und RAD 12/21
U1SB3.py
from math import *
from time import sleep
from tiplotlib import text_at,cls
from ti_system import get_key
#=====
f=input("Frequency? ")
t=input("Time? ")
sound.tone(frequency,time)
0-8000 Hz

```



7. **ctrl+R**を押してプログラムを実行します。プロンプトで、frequencyに440, timeに2を入力します。

そして...おっと! runtime error(ランタイムエラー)! このエラーは、入力関数がサウンド関数では使えない値を提供したため発生します。これは、数値440, 2ではなく文字列("440"と"2")です。float()(10進数の場合)またはint()(整数の場合)を使って、文字列値を数値に変換する必要があります。

```

Python Shell 47/47
Time? 2
Traceback (most recent call last):
  File "<stdin>", line 2, in <module>
  File "C:\Users\jehan\AppData\Roaming\Texas Instruments\TI-Nspire CX CAS Premium Teacher Software\python\doc13\U1SB3.py", line 12, in <module>
  File "python\doc1", line 1615, in tone
TypeError: can't convert 'float' object to str implicitly
>>>|

```

8. プログラムに戻り、各入力関数の前にint()関数を追加します。入力するか、次のメニューからint()関数を見つけることができます。

Menu > Built-ins > Type.(メニュー>組み込み>タイプ)

```

f = int( input( "Frequency? ") )
t = int( input( "Time? ") )

```

すべての括弧について配置に注意します! 括弧を1つ移動する必要があります。各ステートメントの最後に2つの右括弧があることに注意してください。

プログラムを再度実行し440と2を入力すると、TI-Innovator Hubの下部にあるスピーカーから440Hzのトーンが2秒間聞こえます。ctrl+Rを押してシェルでプログラムを再実行し、他の周波数(0~8000Hz)を試しましょう。全部聞こえますか。

```

*U1SB3.py 13/21
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#-----
f=int(input("Frequency? "))
t=int(input("Time? "))
sound.tone(f,t)
|

```

9. Notes(音符)を再生します。

サウンドデバイスは、文字列の名前を使って音符(musical notes)を再生することもできます。

```

sound.note( "A4",2)

```

A4は、周波数440Hzの「ラ」の音で、その音を2秒間鳴らします。

最初の入力関数を変更して、frequency(周波数)ではなくnote(音符)にします。noteは引用符(文字列)であるため、int()関数は必要ありません。

再入力するか、メニューを使ってサウンドコマンドをtoneからnoteに変更します。プログラムを実行するときは、入力はA4(引用符なし)で「ラ」の音符です。

```

*U1SB3.py 10/11
#-----
from ti_hub import *
from math import *
from time import sleep
from ti_plotlib import text_at,cls
from ti_system import get_key
#-----
f=input("Note?")
t=int(input("time?"))
sound.note(f,t)

```